

ABSTRACT

An efficient and adaptive middleware infrastructure called the Event Heap system dynamically coordinates application interactions and communications in a ubiquitous computing environment, e.g., an interactive workspace, having heterogeneous software applications running on various machines and devices across different platforms. Applications exchange events via the Event Heap. Each event is characterized by a set of unordered, named fields. Events are routed by matching certain attributes in the fields. The source and target versions of each field are automatically set when an event is posted or used as a template. The Event Heap system implements a unique combination of features, both intrinsic to tuplespaces and specific to the Event Heap, including content based addressing, support for routing patterns, standard routing fields, limited data persistence, query persistence/registration, transparent communication, self-description, flexible typing, logical/physical centralization, portable client API, at most once per source first-in-first-out ordering, and modular restartability.